CompuScholar, Inc.

Alignment to the K-12 Computer Science Framework

9th - 12th Grade Band

K-12 Computer Science Framework Details:

Standards Link:	K–12-Computer-Science-Framework.pdf

CompuScholar Courses in this Grade Band:

Course Title:	**Digital Savvy**, ISBN 978-0-9887070-8-5 Course Description and Syllabus
Course Title:	**Web Design**, ISBN 978-0-9887070-3-0 Course Description and Syllabus
Course Title:	**Python Programming**, ISBN 978-1-946113-00-9 Course Description and Syllabus
Course Title:	**Computer Science Foundations**, ISBN 978-1-946113-02-3 Course Description and Syllabus
Course Title:	**C# Programming**, ISBN 978-1-946113-01-6 Course Description and Syllabus
Course Title:	**Java Programming**, ISBN 978-1-946113-99-3 Course Description and Syllabus
Course Title:	**Unity Game Programming**, ISBN 978-0-9887070-7-8 Course Description and Syllabus

Note 1: Citation(s) listed may represent a subset of the instances where objectives are met throughout the course.

Note 2: Citation(s) for a "Lesson" refer to the "Lesson Text" elements and associated "Activities" within the course, unless otherwise noted. The "Instructional Video" components are supplements designed to introduce or re-enforce the main lesson concepts, and the Lesson Text contains full details.

K-12 CS Concepts, 9 - 12 Grade Band

K-12 CS SHORTHAND	K-12 CS STATEMENT	COMPUSCHOLAR ALIGNMENT
9–12.Computing Systems.Devices	Computing devices are often integrated with other systems, including biological, mechanical, and social systems. These devices can share data with one another. The usability, dependability, security, and accessibility of these devices, and the systems they are integrated with, are important considerations in their design as they evolve.	Our courses contain introductory chapters on computer hardware, relevant software and emerging trends. Students will learn about relevant security risks and strategies, how data flows between networks, and how devices access network resources. Students will design and create digital artifacts and apps with relevant testing to ensure usability and dependability.

K-12 CS SHORTHAND	K-12 CS STATEMENT	COMPUSCHOLAR ALIGNMENT
9–12.Computing Systems.Hardware and Software	Levels of interaction exist between the hardware, software, and user of a computing system. The most common levels of software that a user interacts with include system software and applications. System software controls the flow of information between hardware components used for input, output, storage, and processing.	Our courses contain lessons that describe hardware components including storage, processing, I/O peripherals and network modules. We discuss operating systems as the intermediary between hardware resources and end user applications. Software applications are categorized by general type.
9–12.Computing Systems. Troubleshooting	Troubleshooting complex problems involves the use of multiple sources when researching, evaluating, and implementing potential solutions. Troubleshooting also relies on experience, such as when people recognize that a problem is similar to one they have seen before or adapt solutions that have worked in the past.	Our courses contain dedicated troubleshooting and debugging information for relevant technology. The programming courses describe how to use a variety of debugging approaches, including code analysis, tracing (logging) and setting breakpoints in a debugger. Best practices and common troubleshooting tips are provided as needed.
9–12.Networks and the Internet.Network Communication and Organization	Network topology is determined, in part, by how many devices can be supported. Each device is assigned an address that uniquely identifies it on the network. The scalability and reliability of the Internet are enabled by the hierarchy and redundancy in networks.	Our courses describe networking components and common network topology. IP addresses, MAC addresses and URLs are introduced for identification of devices and online resources. Network design considerations, including scalability, security and points of failure are addressed.
9–12.Networks and the Internet.Cybersecurity	Network security depends on a combination of hardware, software, and practices that control access to data and systems. The needs of users and the sensitivity of data determine the level of security implemented.	Our courses contain chapters or lessons on relevant security topics such as encryption (including SSL/TLS), online safety and privacy, protecting personal information online, and establishing defenses against mal-ware and viruses.
9–12.Data and Analysis.Collection	Data is constantly collected or generated through automated processes that are not always evident, raising privacy concerns. The different collection methods and tools that are used influence the amount and quality of the data that is observed and recorded.	Our courses teach students to protect information and minimize the personal information shared online. Students learn about their personal digital footprint and understand that digital identities and online actions have long-term or permanent consequences.

K-12 CS SHORTHAND	K-12 CS STATEMENT	COMPUSCHOLAR ALIGNMENT
9–12.Data and Analysis.Storage	Data can be composed of multiple data elements that relate to one another. For example, population data may contain information about age, gender, and height. People make choices about how data elements are organized and where data is stored. These choices affect cost, speed, reliability, accessibility, privacy, and integrity.	Our courses cover representation of data in multiple formats, from simple encoding of data types up through data structures and object definitions with properties. Choices of data type, data structure, or object definitions have pros and cons and may impact the desirable characteristics of the program or the maintainability of the underlying code.
9–12.Data and Analysis.Visualization and Transformation	People transform, generalize, simplify, and present large data sets in different ways to influence how other people interpret and understand the underlying information. Examples include visualization, aggregation, rearrangement, and application of mathematical operations.	Our team projects give students opportunities to research topics, obtain data sets, and produce digital artifacts or apps to visualize and explain the aggregated information. The artifacts or apps can be used to explain or influence opinion about an issue.
9–12.Data and Analysis.Inference and Models	The accuracy of predictions or inferences depends upon the limitations of the computer model and the data the model is built upon. The amount, quality, and diversity of data and the features chosen can affect the quality of a model and ability to understand a system. Predictions or inferences are tested to validate models.	Our courses contain opportunities for student to build models of real-world phenomena and simulate or predict results. Students are encouraged to alter or modify input data to observe the impact on resulting output and verify initial assumptions.
9–12.Algorithms and Programming.Algorithms	People evaluate and select algorithms based on performance, reusability, and ease of implementation. Knowledge of common algorithms improves how people develop software, secure data, and store information.	Our courses contain appropriate analysis and comparison of algorithms, including trade-offs in performance, coding complexity, and accuracy of results. Common sorting and searching algorithms or game-specific AI routines are illustrated.
9–12.Algorithms and Programming.Variables	Data structures are used to manage program complexity. Programmers choose data structures based on functionality, storage, and performance tradeoffs.	Our programming courses cover traditional data structures (arrays, lists, objects, stacks, queues) and the trade-offs involved with each representation. Students will use each data structure in hands-on projects.

K-12 CS SHORTHAND	K-12 CS STATEMENT	COMPUSCHOLAR ALIGNMENT
9–12.Algorithms and Programming.Control	Programmers consider tradeoffs related to implementation, readability, and program performance when selecting and combining control structures.	Our courses discuss trade-offs in design, including selecting between appropriate flow control logic, selecting object models with the best relationships and using functional decomposition to produce easily managed logical blocks. Algorithms are compared for performance and ease of implementation, where appropriate.
9–12.Algorithms and Programming.Modularity	Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose. These modules can be procedures within a program; combinations of data and procedures; or independent, but interrelated, programs. Modules allow for better management of complex tasks.	Our courses cover Object-Oriented Programming (OOP), modular programming with functions, and functional decomposition of complex tasks down to manageable logical blocks.
9–12.Algorithms and Programming.Program Development	Diverse teams can develop programs with a broad impact through careful review and by drawing on the strengths of members in different roles. Design decisions often involve tradeoffs. The development of complex programs is aided by resources such as libraries and tools to edit and manage parts of the program. Systematic analysis is critical for identifying the effects of lingering bugs.	Our courses contain team projects that allow diverse individuals to contribute their strengths to a common project. Students will learn about standard software engineering roles and development life-cycles. Students will write project plans, requirements and design documents, and follow systematic test plans.
9–12.Impacts of Computing.Culture	The design and use of computing technologies and artifacts can improve, worsen, or maintain inequitable access to information and opportunities.	Our courses contain lessons on the impact of global computing, digital accessibility and the concept of net neutrality.
9–12.Impacts of Computing.Social Interactions	Many aspects of society, especially careers, have been affected by the degree of communication afforded by computing. The increased connectivity between people in different cultures and in different career fields has changed the nature and content of many careers.	Our courses contain lessons on the impact of computing in employment markets. We describe essential job-seeking and professional skills and provide specific career exploration activities.

K-12 CS SHORTHAND	K-12 CS STATEMENT	COMPUSCHOLAR ALIGNMENT
9–12.Impacts of Computing.Safety, Law, and Ethics	Laws govern many aspects of computing, such as privacy, data, property, information, and identity. These laws can have beneficial and harmful effects, such as expediting or delaying advancements in computing and protecting or infringing upon people's rights. International differences in laws and ethics have implications for computing.	Our courses cover computing ethics, copyrights, intellectual property, personal privacy and computer security. Relevant laws are discussed in each lesson.

K-12 CS Practices (All Grades)

K-12 CS SHORTHAND	K-12 CS STATEMENT	COMPUSCHOLAR ALIGNMENT
P1.Fostering an Inclusive Computing Culture.1	Include the unique perspectives of others and reflect on one's own perspectives when designing and developing computational products.	Our courses contain lessons on the global impact of computing and digital accessibility. Team projects allow students to collaborate and contribute unique viewpoints to create digital artifacts.
P1.Fostering an Inclusive Computing Culture.2	Address the needs of diverse end users during the design process to produce artifacts with broad accessibility and usability	Our courses contain lessons on digital accessibility and relevant standards. Individual and group projects, where relevant, are required to follow accessibility guidelines.
P1.Fostering an Inclusive Computing Culture.3	Employ self- and peer-advocacy to address bias in interactions, product design, and development methods.	The courses contain guidance on ethical computing, professional behavior and impacts of a global digital environment. Multiple opportunities are available for students to work in teams and provide feedback within the team and to other teams.
P2.Collaborating Around Computing.1	Cultivate working relationships with individuals possessing diverse perspectives, skills, and personalities.	Our courses contain team projects and classroom discussion opportunities that allow diverse individuals to work together to debate, form opinions and produce digital artifacts.
P2.Collaborating Around Computing.2	Create team norms, expectations, and equitable workloads to increase efficiency and effectiveness.	Team projects and related lessons instruct students on creating an appropriate scope, timeline, and executing projects in phases. Team members will work efficiently together or in parallel to meet self-directed project milestones.

K-12 CS SHORTHAND	K-12 CS STATEMENT	COMPUSCHOLAR ALIGNMENT
P2.Collaborating Around Computing.3	Solicit and incorporate feedback from, and provide constructive feedback to, team members and other stakeholders.	Team projects contain testing, evaluation and feedback phases. Peer and professional or teacher review is used to generate actionable suggestions for improvement of digital artifacts.
P2.Collaborating Around Computing.4	Evaluate and select technological tools that can be used to collaborate on a project.	Students are exposed to a variety of productivity tools and will use collaborative services and technology as appropriate during team projects.
P3.Recognizing and Defining Computational Problems.1	Identify complex, interdisciplinary, real-world problems that can be solved computationally.	Courses, where appropriate, contain lessons on modeling real-world problems in a variety of subject areas (e.g. physics, space exploration, monitoring and interpretation of website behavior).
P3.Recognizing and Defining Computational Problems.2	Decompose complex real-world problems into manageable subproblems that could integrate existing solutions or procedures	Courses, where appropriate, contain lessons on algorithms, decomposition of tasks into smaller functions, and the application of artificial intelligence to complement and extend human functionality.
P3.Recognizing and Defining Computational Problems.3	Evaluate whether it is appropriate and feasible to solve a problem computationally.	Courses, where appropriate, contain lessons on the limitations and implementation challenges of algorithms. This includes discussion of whether parts of a complex problem are best handled by an algorithm or AI instead of a human.
P4.Developing and Using Abstractions.1	Extract common features from a set of interrelated processes or complex phenomena.	Courses, where appropriate, evaluate situations to identify algorithmic applications, functions, and repeatable patterns to represent part or all of a task.
P4.Developing and Using Abstractions.2	Evaluate existing technological functionalities and incorporate them into new designs.	In team projects, students will research selected topics and select appropriate technical tools (e.g. word processors, spreadsheets, databases, websites, apps) to visualize and present information to new audiences.
P4.Developing and Using Abstractions.3	Create modules and develop points of interaction that can apply to multiple situations and reduce complexity.	Courses, where appropriate, introduce functions and decomposition of complex problems into more manageable tasks.
P4.Developing and Using Abstractions.4	Model phenomena and processes and simulate systems to understand and evaluate potential outcomes.	Courses, where appropriate, introduce modeling of physical and other phenomena and simulations to predict outcomes across a variety of subject areas.

K-12 CS SHORTHAND	K-12 CS STATEMENT	COMPUSCHOLAR ALIGNMENT
P5.Creating Computational Artifacts.1	Plan the development of a computational artifact using an iterative process that includes reflection on and modification of the plan, taking into account key features, time and resource constraints, and user expectations.	Students will learn about project management tools such as project plans, requirements & design documents, development lifecycles (e.g. classic waterfall), and apply those tools and skills in group or individual projects.
P5.Creating Computational Artifacts.2	Create a computational artifact for practical intent, personal expression, or to address a societal issue.	Courses include team or individual projects where students will select a topic or issue of interest, research and gather data, and create one or more digital artifacts to summarize or interpret the results.
P5.Creating Computational Artifacts.3	Modify an existing artifact to improve or customize it.	Students are frequently given "starter" projects as the basis for completing more complex activities. They are also given opportunities to examine output from other teams, provide feedback, and incorporate feedback to make improvements.
P6.Testing and Refining Computational Artifacts.1	Systematically test computational artifacts by considering all scenarios and using test cases.	Courses contain dedicated chapters and lessons on debugging, test plans and troubleshooting skills. Team projects include iterative testing and fixing phases to address bug reports and feedback.
P6.Testing and Refining Computational Artifacts.2	Identify and fix errors using a systematic process.	Courses contain dedicated chapters and lessons that describe appropriate procedures for finding and fixing problems or bugs. Test plans and checklists are introduced as useful tools to systematically evaluate a digital artifact.
P6.Testing and Refining Computational Artifacts.3	Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability, and accessibility.	Team projects include iterative test and feedback phases. Students will incorporate test reports and feedback from peers and professionals to improve their digital artifacts.
P7.Communicating About Computing.1	Select, organize, and interpret large data sets from multiple sources to support a claim.	Team projects ask students to research a topic of interest, obtain data, incorporate that data into an appropriate digital expression, and interpret the results for a wider audience.
P7.Communicating About Computing.2	Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose.	The courses describe how to create requirements and design documents, storyboards, flowcharts, end user documents with appropriate languages as

K-12 CS SHORTHAND	K-12 CS STATEMENT	COMPUSCHOLAR ALIGNMENT
P7.Communicating About Computing.3	Articulate ideas responsibly by observing intellectual property rights and giving appropriate attribution.	The courses contain dedicated lessons on computing ethics, intellectual property, copyrights, and proper citation. Students are required to adhere to these standards throughout each course.